

University of Groningen

## Constructing Universally Rigid Tensegrity Frameworks with Application in Multi-Agent Formation Control

Yang, Qingkai; Cao, Ming; Fang, Hao; Chen, Jie

*Published in:*  
IEEE Transactions on Automatic Control

*DOI:*  
[10.1109/TAC.2018.2829687](https://doi.org/10.1109/TAC.2018.2829687)

**IMPORTANT NOTE:** You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

*Document Version*  
Final author's version (accepted by publisher, after peer review)

*Publication date:*  
2019

[Link to publication in University of Groningen/UMCG research database](#)

### *Citation for published version (APA):*

Yang, Q., Cao, M., Fang, H., & Chen, J. (2019). Constructing Universally Rigid Tensegrity Frameworks with Application in Multi-Agent Formation Control. *IEEE Transactions on Automatic Control*, 64(1), 381-388. <https://doi.org/10.1109/TAC.2018.2829687>

### **Copyright**

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

### **Take-down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

# Constructing universally rigid tensegrity frameworks with application in multi-agent formation control

Qingkai Yang *Student Member, IEEE*, Ming Cao, *Senior Member, IEEE*, Hao Fang, *Member, IEEE*, and Jie Chen, *Senior Member, IEEE*

**Abstract**—Rigidity graph theory has found broad applications in engineering, architecture, biology and chemistry, while systematic and computationally tractable construction of rigid frameworks is still a challenging task. In this paper, starting from any given configuration in general positions, we show how to construct a universally rigid tensegrity framework by looking into the kernel of the tensegrity framework's stress matrix. As one application, we show how to stabilize a formation of mobile agents by assigning a universally rigid virtual tensegrity framework for the formation and then design distributed controllers based on the forces determined by the stresses of the edges. Such formation controllers are especially useful when one needs to satisfy formation constraints in the form of strict upper or lower bounds on inter-agent distances arising from tethered robots.

**Index terms**— Universal rigidity, stress matrix, tensegrity framework, formation control, multi-agent system

## I. INTRODUCTION

Rigidity graph theory has always been playing a key role in solving topology related problems in various fields, e.g. the formation control problem of multi-agent systems [1–3], the geometric analysis of molecular models in bio-chemistry [4], and the localization of wireless sensor networks [5]. To characterize the rigidity, the commonly used bar-joint frameworks have been extensively studied. Intuitively speaking, (local) rigidity determines the uniqueness of the framework up to congruence, which implies the smooth motions of the framework are those corresponding to the combinations of translation, rotation and reflection of the whole framework. When we require the uniqueness of the framework in the whole space of some specific dimension, the framework needs to be globally rigid. Furthermore, a stronger notion describing that a framework is uniquely determined up to congruence in any higher dimensional space is universal rigidity.

A challenging problem concerned with rigidity is to determine whether a given framework is rigid (resp. globally rigid and universally rigid). It is already well known that the local rigidity can be guaranteed if the rank of the corresponding rigidity matrix exceeds some bound. However, for global rigidity and universal rigidity, the problem is shown to be NP-hard even in the lower dimensional space  $\mathbb{E}^1$  [6]. Then, to make the problem more tractable, researchers concentrate on the frameworks with a *generic configuration*, where the nodal coordinates are algebraically independent over the rationals. Some exploring efforts along this line have been made in [7–9], where the sufficient conditions for a framework of generic configurations to be globally/universally rigid have been given, and the necessity

of these conditions is later proved in [10]. Recently, a new kind of universal rigidity called ‘iterative universal rigidity’ has been addressed in [11], where the analysis of the rigidity of the framework is decomposed into a sequence of affine sets in the configuration space. One weaker condition compared with being generic for the configuration of the framework is *general*, which is preferable in practice since it is checkable in polynomial time, while not for the generic configuration [12]. It has been verified that the genericity assumption of the configuration to ensure universal rigidity can be relaxed to the situation of general positions in [13]. In addition, the universal rigidity of one dimensional frameworks in general positions with a complete bipartite underlying graph is fully discussed in [14].

Of particular theoretical and practical interest is a class of frameworks called *tensegrity frameworks*, whose edges consist of three different types of *members*: *cables*, only allowed to become shorter; *struts*, only allowed to become longer, and *bars*, constrained to maintain a fixed length [15]. In recent years, they have found broad applications in engineering, architecture, biology and arts because of their superior features, such as deployability, deformability and robustness [16, 17]. However, little research has been conducted to construct universally rigid tensegrity frameworks from some given geometric shape described by the overall configuration. One of the well-established findings is that for a framework in the shape of convex polygons in the plane, the universal rigidity can be obtained if the boundary and the interior members are respectively set to be cables and struts [18]. For a class of so called Grünbaum frameworks, the construction for universally rigid tensegrity frameworks in two and three dimensions is studied in [19], in which the approach strongly relies on the computation of the convex hull. If the underlying graph of the framework is given beforehand, a purification based algorithm is designed to compute the corresponding stresses for the  $(d + 1)$ -lateral frameworks in general positions [12]. In the case that only the configuration is known, to create a universally rigid tensegrity framework, the underlying graph together with the type of members are determined through the algorithm in [20], while it is highly likely to result in a complete graph.

For control engineers, the issue of identifying and designing rigid frameworks are particularly relevant for the formation control problem of teams of mobile agents. The stable tensegrity frameworks, due to their strong robustness and clear physical interpretation, have been used as the virtual multi-agent framework to understand better the behavior of the agents around the equilibrium corresponding to the prescribed formation shape. But the existing results are restricted to one-dimensional tensegrity frameworks [21] or special classes of tensegrity frameworks, e.g. *cross-tensegrities* that are rectangles with four boarder cables and two crossing struts [22, 23].

Motivated by the recent advances in rigidity graph theory, it is the goal of this paper to first design an algorithm to construct a universally rigid tensegrity framework for any given configurations in general positions. Taking into account the engineering and theoretical concerns that sparse frameworks are more desirable because of extendibility, complexity and computation, we will focus on building tensegrity frameworks with fewer members through constructing

This work was supported by Projects of Major International (Regional) Joint Research Program NSFC (Grant no. 61720106011), NSFC (Grant no. 61621063, 6157306261673058), Beijing Advanced Innovation Center for Intelligent Robots and Systems (Beijing Institute of Technology), Key Laboratory of Biomimetic Robots and Systems (Beijing Institute of Technology), Ministry of Education, Beijing, 100081, China.

Q. Yang and M. Cao are with the Faculty of Science and Engineering, University of Groningen, Groningen 9747 AG, the Netherlands.

Q. Yang, H. Fang and J. Chen are with the School of Automation, Beijing Institute of Technology, Beijing, 100081, P. R. China. e-mail: fangh@bit.edu.cn

stress matrices with close to maximal allowed number of zeros. We then use the constructed universally rigid tensegrity framework for multi-agent formation control. Distributed control laws are designed to stabilize formations when their topologies are implemented using the virtual tensegrity framework. So the main contribution of this paper is two-fold. First, we develop a systematic algorithm to construct universally rigid tensegrity frameworks. Such algorithms have not been reported in the literature. Second, we apply the virtual tensegrity frameworks to distributed formation control, which therefore enables us to have a clear intuitive estimate of the domain of attraction around the system's equilibrium. This application is particularly useful for the emerging cooperative control of tethered robots, where the challenging formation constraints, such as the strict maximum or minimum inter-agent distances, could be incorporated in the virtual tensegrity framework.

The rest of the paper is organized as follows. In Section II we review some basic concepts of rigidity and sufficient conditions for universally rigid tensegrity frameworks. We propose our algorithm to construct universally rigid tensegrity frameworks for any given configuration in general position in Section III. The application for formation control is discussed in Section IV. Some illustrative examples are presented in Section V.

## II. BACKGROUND

We follow the convention in [8, 11] to present a brief overview of rigidity graph theory and relevant knowledge. Let  $\mathcal{V} = \{1, 2, \dots, n\}$  and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  be, respectively, the vertex set and the edge set of an undirected graph  $\mathcal{G}$  representing the neighboring relationships between the  $n$  vertices. There is an edge  $(i, j)$  if and only if vertex  $i$  and  $j$  are neighbors of each other. By assigning an arbitrary orientation to  $\mathcal{G}$ , the incidence matrix  $H = [h_{ij}] \in \mathbb{R}^{|\mathcal{E}| \times n}$  is defined by

$$h_{ij} = \begin{cases} 1, & \text{ith edge enters node } j, \\ -1, & \text{ith edge leaves node } j, \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where  $|\mathcal{E}|$  represents the cardinality of the edge set  $\mathcal{E}$ . A *configuration* is a finite collection of  $n$  labeled points in  $d$ -dimensional Euclidean space  $\mathbb{E}^d$ , denoted by  $q = [q_1, \dots, q_n]$ . A *framework*  $(\mathcal{G}, q)$  is obtained by embedding an undirected graph  $\mathcal{G}$  in  $\mathbb{E}^d$  together with its corresponding configuration  $q$ , where the graph is finite and without loops or multiple edges.

Given a framework  $(\mathcal{G}, q)$  in  $\mathbb{R}^d$ , if there exists another framework  $(\mathcal{G}, p)$  in  $\mathbb{R}^d$  such that  $\|p_i - p_j\| = \|q_i - q_j\|, \forall (i, j) \in \mathcal{E}$ , then we say that  $(\mathcal{G}, p)$  is *equivalent* to  $(\mathcal{G}, q)$ . Furthermore, they are *congruent* if  $\|p_i - p_j\| = \|q_i - q_j\|, \forall i, j \in \mathcal{V}$ . With these concepts, we say that a framework  $(\mathcal{G}, q)$  in  $\mathbb{R}^d$  is

- *(locally) rigid*, if all the frameworks  $(\mathcal{G}, p)$  in  $\mathbb{R}^d$  equivalent to  $(\mathcal{G}, q)$  and sufficiently close to  $(\mathcal{G}, q)$  are congruent to  $(\mathcal{G}, q)$ ;
- *globally rigid*, if all the frameworks  $(\mathcal{G}, p)$  in  $\mathbb{R}^d$  equivalent to  $(\mathcal{G}, q)$  are congruent to  $(\mathcal{G}, q)$ ;
- *universally rigid*, if all the frameworks  $(\mathcal{G}, p)$  in any  $\mathbb{R}^D \supset \mathbb{R}^d$  equivalent to  $(\mathcal{G}, q)$  are congruent to  $(\mathcal{G}, q)$ .

To characterise the universal rigidity of a tensegrity framework, we also employ the concept of stress. For each edge  $(i, j)$  of a tensegrity framework  $(\mathcal{G}, q)$ , we assign a scalar  $\omega_{ij} = \omega_{ji}$ , and use  $\omega \in \mathbb{R}^{|\mathcal{E}|}$  to denote the concatenated vector  $\omega = (\dots, \omega_{ij}, \dots)^T$ . Then  $\omega$  is called a *stress* of  $(\mathcal{G}, q)$ . If further, each  $\omega_{ij}$  satisfies  $\omega_{ij} \geq 0$  whenever  $(i, j)$  is a cable and  $\omega_{ij} \leq 0$  whenever  $(i, j)$  is a strut, then  $\omega$  is said to be a *proper stress*. Note that for a stress to be proper, there is no restriction on a bar. In physics,  $\omega_{ij}$  is interpreted as the axial force per unit length along the edge  $(i, j)$ . Let  $q^*$  be

a given configuration. Then we call that  $\omega$  an *equilibrium stress* of  $(\mathcal{G}, q^*)$  if it is a solution to the equation set

$$\sum_{j \in \mathcal{N}_i} \omega_{ij} (q_j^* - q_i^*) = \mathbf{0}, \quad i = 1, \dots, n, \quad (2)$$

where  $\mathcal{N}_i$  is the set of adjacent vertices of  $i$  in  $\mathcal{G}$ . Given  $\omega$ , the associated *stress matrix*  $\Omega \in \mathbb{R}^{n \times n}$  is defined by letting  $\Omega_{ij} = -\omega_{ij}$  for  $i \neq j$  and  $\Omega_{ii} = \sum_{j \neq i} \omega_{ij}$  for  $i = 1, \dots, n$ .

In this paper, we consider that the given configuration  $q^* \in \mathbb{R}^{d \times n}$  is *general*, i.e., no  $d+1$  points of  $q_1^*, \dots, q_n^*$  are affinely dependent. We introduce here the following lemma to be used later.

**Lemma 1.** [11] *Let  $(\mathcal{G}, q)$  be a tensegrity framework whose affine span of  $q$  is  $\mathbb{R}^d$ , with an equilibrium stress  $\omega$  and stress matrix  $\Omega$ . Suppose further that*

- 1)  $\Omega$  is positive semi-definite,
- 2) the rank of  $\Omega$  is  $n - d - 1$ ,
- 3) and the configuration  $q$  is in general position,

*then  $(\mathcal{G}, q)$  is universally rigid.*

With the knowledge about rigidity properties at hand, now we are ready to propose our algorithm to construct universally rigid tensegrity frameworks given their shapes specified by  $q^*$ .

## III. CONSTRUCTING UNIVERSALLY RIGID TENSEGRITY FRAMEWORKS

We first provide the steps of the algorithm in detail and then show the constructed tensegrity frameworks are in general close to minimal by giving an upper bound of the numbers of their members.

### A. Algorithm

We assume the given configuration  $q^* \in \mathbb{R}^{d \times n}$  is in general position. Define the extended configuration matrix  $Q^* \triangleq [(q^*)^T, \mathbf{1}_n]^T \in \mathbb{R}^{(d+1) \times n}$ . Then, it follows that every  $(d+1) \times (d+1)$  submatrix of  $Q^*$  has full rank. From the definition of the stress matrix  $\Omega$  and (2), one can check that matrix  $\Omega$  always lives in the null space of  $Q^*$ , i.e.

$$Q^* \Omega = \mathbf{0}_{(d+1) \times n}. \quad (3)$$

Given  $q^*$ , the key component of the algorithm is to determine matrix  $\Omega$ , which determines in turn which two nodes are connected together with their nonzero stress. Obviously, such  $\Omega$  is in general not unique and naturally we want to obtain an  $\Omega$  with more zeros which leads to fewer members and thus lower complexity. Towards this end, we convert our problem into the *sparse null space problem* first considered in [24], namely, given a matrix  $A$ , to find a sparse matrix  $B$  such that  $B$  is full rank and its column span is  $\text{null}(A)$  [25].

In view of Lemmas 1, to obtain a universally rigid tensegrity framework, matrix  $\Omega$  is required to be positive semi-definite with rank  $n - d - 1$ . However, since  $\Omega$  in (3) is not full rank, we cannot directly solve the sparse null space problem. Instead, we try to construct a column full-rank matrix  $D \in \mathbb{R}^{n \times (n-d-1)}$  such that

$$Q^* D = \mathbf{0}_{(d+1) \times (n-d-1)}, \quad (4)$$

where  $D$  is a *Gale matrix* of  $q^*$  [13]. If indeed such a  $D$  can be constructed, it must be true that

$$Q^* D D^T = \mathbf{0}_{(d+1) \times (n-d-1)} D^T = \mathbf{0}_{(d+1) \times n}, \quad (5)$$

and hence the matrix  $D D^T$  can serve as the stress matrix  $\Omega$ . So the construction of an  $\Omega$  is equivalent to the design of such a sparse  $D$ . In addition, for computational efficiency, we make an even stronger requirement that  $\Omega$  is in its *band form*, whose non-zero entries are

confined to be in a diagonal band containing the main diagonal. Now we present our 5-step algorithm to construct the universally rigid framework with the stress matrix  $\Omega$ , which is inspired by the classical “turning back” method for computing the sparse null space basis [26].

Step 1: Arrange matrix

$$Q^* = [(q^*)^T, \mathbf{1}_n]^T \in \mathbb{R}^{(d+1) \times n}.$$

Step 2: Locate the nonzero elements of  $D$ . We first find the smallest  $k_1 > 0$  such that  $Q^*$ 's columns with the indices  $d+2, d+1, \dots, d+2-k_1$  are linearly dependent. We then set the nonzero elements of  $D$ 's first column to be located at the positions  $d+2-k_1$  through  $d+2$ . Then, in order to record the positions of the nonzero elements of the second column of  $D$ , we find the smallest  $k_2 > 0$  such that those columns with the indices  $d+3, d+2, \dots, d+3-k_2$ , excluding  $d+2-k_1$ , of  $Q^*$  are linearly dependent. Again, the indices correspond to the nonzero elements' positions of  $D$ 's second column. We repeat this procedure until we have determined the positions of the nonzero elements of the last column of  $D$ . Note that the configuration  $q^*$  is in general position, there are exactly  $d+2$  nonzero elements in each column of  $D$ , yielding

$$D = \begin{pmatrix} 1 & 2 & \cdots & n-d-1 \\ * & 0 & \cdots & 0 \\ * & * & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & 0 \\ 0 & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & * \end{pmatrix} \begin{matrix} 1 \\ 2 \\ \vdots \\ d+2 \\ d+3 \\ \vdots \\ n \end{matrix} \quad (6)$$

where ‘\*’ indicates a nonzero element.

Step 3: Now we compute the values of the nonzero elements of  $D$  by solving the following equation

$$Q^* D = \mathbf{0}_{(d+1) \times (n-d-1)}, \quad (7)$$

which is underdetermined since it is a set of  $d+1$  linear equations with  $d+2$  unknowns. Hence, we can always find a set of nonzero elements of  $D$  and thus fully determine  $D$ . In addition, it is easy to check that the constructed  $D$  is always column full-rank.

Step 4: The stress matrix is then the positive semidefinite matrix  $\Omega = DD^T$  whose rank is  $n-d-1$ . The obtained  $\Omega$  is a square, symmetric, band matrix with one upper triangular and one lower triangular  $(n-d-2)$ -dimensional submatrix in its upper right and lower left corners respectively. We write  $\Omega$  in the notation below where all the determined zero elements are denoted by zero and all the other elements that may or may not be zero are denoted by ‘×’:

$$\Omega = \begin{bmatrix} \times & \cdots & \times & 0 & \cdots & \cdots & 0 \\ \times & \cdots & \cdots & \times & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & & \ddots & \vdots \\ \times & \cdots & \cdots & \times & \cdots & \times & 0 \\ \times & \cdots & \cdots & \times & \cdots & \cdots & \times \\ \vdots & & & \vdots & & & \vdots \\ 0 & \times & \cdots & \times & \cdots & \cdots & \times \\ \vdots & \ddots & & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & \times & \cdots & \cdots & \times \\ 0 & \cdots & \cdots & 0 & \times & \cdots & \times \end{bmatrix}_{n \times n} \quad (8)$$

Step 5: According to  $\Omega$ , assign cables and struts to the  $n$ -node framework: For each  $\Omega_{ij} < 0$ , we assign a cable between nodes  $i$  and  $j$  and for each  $\Omega_{ii} > 0$ , we assign a strut between nodes  $i$  and  $j$ . The stresses of the assigned cables and struts are  $\omega_{ij} = -\Omega_{ij}$ . The desired tensegrity framework is then obtained.

Now we prove that the constructed tensegrity framework is indeed universally rigid.

**Theorem 1.** *Given a configuration  $q^*$  in general position, the proposed 5-step algorithm returns a universally rigid tensegrity framework with the geometric shape prescribed by  $q^*$ .*

*Proof of Theorem 1.* The constructed tensegrity framework has the stress matrix  $\Omega$  which, as shown in step 4, is positive semi-definite and has rank  $n-d-1$ . So the conditions (1) in Lemma 1 is satisfied. Recalling that the given configuration  $q^*$  is in general position, so the obtained tensegrity framework is again universally rigid because of Lemma 1.  $\square$

**Remark 1.** *In step 4, the stress matrix  $\Omega$  can be more generally determined by  $\Omega = D\Psi D^T$ , where  $\Psi$  is a nonsingular,  $(n-d-1) \times (n-d-1)$ , symmetric matrix, used to adjust the magnitudes of the stresses of the members of the constructed tensegrity framework. In this paper, we set  $\Psi = I_{n-d-1}$  for simplicity.*

In practice, one usually prefers fewer edges to reduce the complexity, if possible, in a rigid framework. Hence, a natural question to ask is whether the tensegrity framework obtained by the proposed algorithm is indeed structurally simple. To address this question, we construct an upper bound of the number of members of the obtained tensegrity framework, namely  $|\mathcal{E}|$  of  $(\mathcal{G}, q^*)$ , which is roughly  $nd$  when  $n$  is big.

#### B. Upper bound of $|\mathcal{E}|$

We look into the number of nonzero elements of  $\Omega$  in (8). The densest  $\Omega$ , namely that contains the largest possible number of nonzero elements, appears when all the elements denoted by ‘×’ are nonzero. Then the number of off-diagonal zero elements in  $\Omega = DD^T$  is

$$2((n-d-2) + (n-d-3) + \cdots + 1) = (n-d-1)(n-d-2),$$

or equivalently the number of off-diagonal nonzero elements in  $\Omega$  is

$$n(n-1) - (n-d-1)(n-d-2) = (d+1)(2n-d-2),$$

which, when divided by 2, is exactly the number of members to be inserted into the tensegrity framework in view of Step 5 of the algorithm. Therefore, this densest  $\Omega$  gives an upper bound  $(d+1)(n - \frac{d+2}{2})$ .

So we have proved the following theorem.

**Theorem 2.** *The number of members of the constructed tensegrity framework is upper bounded by*

$$|\mathcal{E}| \leq (d+1) \left( n - \frac{d+2}{2} \right). \quad (9)$$

**Remark 2.** *One necessary condition for a tensegrity framework in generic configurations (with  $n$  larger than  $d+1$ ) to be globally rigid or universally rigid is that it has to be rigid with at least one self-stress state, which in turn implies that it needs to have at least  $nd - d(d+1)/2 + 1$  members. This is the lower bound of the number of members required to construct a globally rigid or universally rigid tensegrity framework in generic configurations. The lower bound differs from the constructed upper bounded roughly by  $n$  when  $n$  is big.*



**Remark 3.** The upper bound of the number of members presented in (9) corresponds to the number of members of universally rigid tensegrity frameworks constructed on  $(d+1)$ -tree graphs [12]. The construction of universally rigid Grünbaum frameworks in generic configurations with the minimal number of edges in 2D and 3D are investigated in [19], where it is shown that the two-dimensional Grünbaum frameworks in nongeneric configurations are also universally rigid. The problem on how to compute the stress matrix has also been considered in [20], which, however, most likely yields stress matrices without any zero elements. In comparison, our algorithm in general always returns a stress matrix with close to maximal allowed number of zeros.

In the next section, we show how universally rigid tensegrity frameworks can be used as virtual structures to help the design of distributed formation controllers for teams of mobile agents.

#### IV. FORMATION STABILIZATION

##### A. Formation Control Problem

We consider a group of  $n$  mobile agents, each of which is modeled by a kinematic point

$$\dot{q}_i = u_i, \quad i = 1, 2, \dots, n, \quad (10)$$

where  $q_i \in \mathbb{R}^d$  represents agent  $i$ 's position and  $u_i \in \mathbb{R}^d$  is its control input. The neighbor relationships will be designed and characterized by an undirected graph  $\mathcal{G}$  with the vertex set  $\mathcal{V}$  and the edge set  $\mathcal{E}$ .

The *formation stabilization problem* is that given a desired configuration  $q^* = [q_1^*, \dots, q_n^*] \in \mathbb{R}^{d \times n}$  for this team of  $n$  agents (10), design the neighbor relationship graph  $\mathcal{G}$  for the team and correspondingly, for each agent  $i = 1, \dots, n$ , design distributed control laws  $u_i(q_i - q_j, q_i^* - q_j^*)$ ,  $j \in \mathcal{N}_i$ , such that the agents' positions are driven to the target set

$$\mathcal{T} = \{q \in \mathbb{R}^{dn} \mid q_i - q_j = q_i^* - q_j^*, \quad \forall (i, j) \in \mathcal{E}\}. \quad (11)$$

Note that here  $q^*$  is given in an arbitrary coordinate system of choice. When each agent has its own coordinate system that may differ from each other,  $q^*$  is then given to each agent in its own coordinate system, in which case different agents' given  $q^*$  differ up to some congruent transformation of translation and rotation.

Ample previous work has discussed how to solve this problem locally when  $(\mathcal{G}, q^*)$  is rigid and the controllers are derived by attaching virtual springs to the agents. To make the controllers simpler,  $\mathcal{G}$  is usually required to contain as few edges as possible. In what follows, we use the virtual tensegrity framework constructed in Section III to describe the necessary sensings between the agents and the resulted distributed controllers whose gains are derived from the stresses.

##### B. Controller design and stability analysis

We use the universally rigid tensegrity framework constructed previously to determine which agents need to sense which other agents. To be specific, from the given  $q^*$ , we run the 5-step algorithm and obtain a framework  $(\mathcal{G}, q^*)$ . Then the underlying graph  $\mathcal{G}$  is used to represent the sensing graph of the  $n$ -agent team.

To calculate the virtual forces utilizing the stresses, we set the rest length of each edge to be

$$l_{ij} = \gamma_{ij} \|r_{ij}^*\| = \begin{cases} \gamma_{ij}^c \|r_{ij}^*\| & \text{if } \omega_{ij} > 0, \\ \gamma_{ij}^s \|r_{ij}^*\| & \text{if } \omega_{ij} < 0, \end{cases} \quad (12)$$

where  $\gamma_{ij}^c \in (0, 1)$  and  $\gamma_{ij}^s \in (1, +\infty)$  are constants as design parameters, and  $r_{ij}^*$  is the prescribed relative position of agent  $j$  with

respect to agent  $i$ , i.e.,  $r_{ij}^* = q_j^* - q_i^*$ . Then the formation control gain  $k_{ij}$  for agent  $i$  with respect to its neighbor  $j$  is given by

$$k_{ij} = \frac{\omega_{ij}}{1 - \gamma_{ij}}, \quad \forall (i, j) \in \mathcal{E}. \quad (13)$$

For convenience in notation, we define the auxiliary variable

$$z_i = q_i - q_i^*. \quad (14)$$

Let  $z \in \mathbb{R}^{dn}$  be the vector obtained by stacking all the  $z_i$  together.

Now, we define the set  $\mathcal{D}(0)$  as

$$\mathcal{D}(0) \triangleq \{z(0) \in \mathbb{R}^{dn} \mid \|z_i(0) - z_j(0)\| < \text{sign}(\omega_{ij})(\|r_{ij}^*\| - l_{ij}), \quad \forall (i, j) \in \mathcal{E}\}. \quad (15)$$

To proceed, we assume all the agents are initially located in the set  $\mathcal{D}^\epsilon(0)$  defined as

$$\mathcal{D}^\epsilon(0) \triangleq \{z(0) \in \mathbb{R}^{dn} \mid \|z_i(0) - z_j(0)\| \leq \text{sign}(\omega_{ij})(\|r_{ij}^*\| - l_{ij}) - \epsilon, \quad \forall (i, j) \in \mathcal{E}\}, \quad (16)$$

where  $\epsilon$  is a small positive number.

The potential function for agent  $i$  is defined as

$$P_i(z(t)) \triangleq \sum_{\omega_{ij} < 0} \frac{k_{ij} [(l_{ij} - \|r_{ij}^*\|)^\alpha - \rho_{ij}^\alpha]^\beta}{\rho_{ij}} + \sum_{\omega_{ij} > 0} \frac{k_{ij} [( \|r_{ij}^*\| - l_{ij})^\alpha - \varrho_{ij}^\alpha]^\beta}{\varrho_{ij}}, \quad (17)$$

where

$$\begin{aligned} \rho_{ij} &= l_{ij} - \|r_{ij}^*\| - \|r_{ij}(t) - r_{ij}^*\|, \\ \varrho_{ij} &= \|r_{ij}^*\| - l_{ij} - \|r_{ij}(t) - r_{ij}^*\|, \end{aligned} \quad (18)$$

with  $r_{ij} = q_j - q_i$  being the relative position between agents  $i$  and  $j$ .  $\alpha > 2$  and  $0 < \beta < 1$  are the positive exponents. Note that in the set  $\mathcal{D}^\epsilon(0)$ ,  $\rho_{ij}$  and  $\varrho_{ij}$  are both small positive numbers. The parameters  $k_{ij}$  are chosen such that

$$P_0 = \begin{cases} k_{ij} (\|r_{ij}^*\| - l_{ij})^{\alpha\beta}, & \text{if } \omega_{ij} > 0, \\ k_{ij} (l_{ij} - \|r_{ij}^*\|)^{\alpha\beta}, & \text{if } \omega_{ij} < 0, \end{cases} \quad (19)$$

where  $P_0$  is an arbitrary positive pre-defined constant. All the parameters  $k_{ij}$  would be determined if any one of them is decided with given  $\alpha$  and  $\beta$ .

Correspondingly, the potential function for the whole system is given by

$$P(z(t)) = \frac{1}{2} \sum_{i=1}^n P_i(z(t)) \quad (20)$$

The control input of agent  $i$  is

$$\dot{q}_i = \dot{z}_i = u_i = -\nabla_{z_i} P_i(z(t)) \quad (21)$$

**Proposition 1.** For any given initial position  $q(0) \in \mathcal{D}^\epsilon(0)$ , the set  $\mathcal{D}(0)$  is invariant to the system (10) under the control law (21).

Before presenting the proof for Proposition 1, we first analyse the properties of the potential function (17). Consider the numerators of (17), where  $\delta_s$  (resp.  $\delta_c$ ) are regarded as independent variables. Define function  $f(x)$  as

$$f(x) = k_f (c_f^\alpha - x^\alpha)^\beta > 0, \quad k_f > 0, x \in (0, c_f], \quad (22)$$

which coincides with the form of the numerators of the potential function. The first order derivative of  $f(x)$  satisfies

$$f'(x) = -\alpha\beta k_f x^{\alpha-1} (c_f^\alpha - x^\alpha)^\beta < 0, \quad x \in (0, c_f], \quad (23)$$

which implies that the function  $f(x)$  is monotonically decreasing with respect to  $x$ . Besides this, for sufficiently small  $x_s$ ,  $f'(x_s) \rightarrow 0$ , due to  $x_s^{\alpha-1} \rightarrow 0$ , when  $\alpha > 2$ .

Look at the second order derivative of  $f(x)$ ,

$$f''(x) = -\alpha(\alpha-1)\beta k_f x^{\alpha-2}(c_f^\alpha - x^\alpha)^{\beta-1} + \alpha^2\beta(\beta-1)k_f x^{2(\alpha-1)}(c_f^\alpha - x^\alpha)^{\beta-2}. \quad (24)$$

It follows from  $\alpha > 2$  and  $0 < \beta < 1$  that  $f'' < 0$ ,  $\forall x \in (0, c_f]$ . Together with the fact that  $f'(x) < 0$ , it implies that the decreasing rate of  $f(x)$  increases as  $x$  grows.

Consider

$$f(x) = f(0) + \int_0^x f'(s)ds = f(0) - \int_0^x |f'(s)|ds, \quad x \in (0, c_f]. \quad (25)$$

It can be seen that  $f'(x)$  drops rapidly, when the value of  $x$  increases.

Based on  $f(x)$ , together with (17), we introduce the function  $P_f(x)$  as

$$P_f(x) = \frac{f(x)}{x}, \quad x \in (0, c_f]. \quad (26)$$

The first order derivatives of  $P_f(x)$  satisfies

$$P_f'(x) < 0. \quad (27)$$

It can be seen that  $P_f(x)$  has the same monotonicity with the potential function in (17) for a single edge. Hence, the potential energy (17) is monotonically decreasing with respect to  $\delta_s$  and  $\delta_c$ . This property implies that the closer the edges approach their rest lengths, the higher their potential energy will become.

Now, we give the proof of Proposition 1, which is partially motivated by [27].

*Proof of Proposition 1.* Recall the control input (21) for agent  $i$

$$u_i = -\nabla_{z_i} P_i(z(t)). \quad (28)$$

Hence the time derivative of  $P_i(t)$  satisfies,

$$\begin{aligned} \dot{P}_i(z(t)) &= \nabla_{z_i} P_i(z(t))^T \dot{z}_i(t) \\ &= -[\nabla_{z_i} P_i(z(t))]^T [\nabla_{z_i} P_i(z(t))] \leq 0, \end{aligned} \quad (29)$$

which implies that the potential energy associated with agent  $i$  is not increasing. However, the potential energy stored in some edge, say  $(i, \nu)$  with negative stress, might increase, even though  $P_i(z(t))$  is not increasing. This means that the length of  $(i, \nu)$  tends to its rest length  $l_{i\nu}$  and the other edges are driven far away from their rest lengths. Since the potential function for every edge defined in (17) is monotonically decreasing, the potential energy of edge  $(i, \nu)$  will increase when  $\|r_{i\nu}\|$  approaches to its rest length, i.e.,  $l_{i\nu}$ . Meanwhile, since the energy generated by the single edge  $(i, \nu)$  is less than that generated by all the edges that connecting agent  $i$ , by letting  $P_i(z(t)) = P_{i\nu}(t)$ , we can get the maximum length for edge  $(i, \nu)$ . Then, it can be checked whether  $\|r_{i\nu}\|$  reaches its rest length  $l_{i\nu}$ .

Now, we consider the length changes of the edges. Since the potential energy  $P_i(z(t))$  is not increasing, we have

$$P_{i\nu}(t) + [P_i(z(t)) - P_{i\nu}(t)] \leq P_i(z(0)), \quad t \geq 0, \quad (30)$$

where  $P_{i\nu}(t)$  denotes the potential energy of edge  $(i, \nu)$  stored at time  $t$ .  $P_i(z(t))$  is the potential energy of all the neighbor edges of agent  $i$ . Obviously, we have  $P_i(z(t)) > P_{i\nu}(t)$ . Then, from (30), it is straightforward to check

$$P_{i\nu}(t) \leq P_i(z(0)) - [P_i(z(t)) - P_{i\nu}(t)] < P_i(z(0)). \quad (31)$$

Hence, to obtain the upper bound of  $\|r_{i\nu}\|$ , it is assumed that

$$P_{i\nu}(t) = P_i(z(0)). \quad (32)$$

According to (27), the potential energy  $P_i(z(t))$  would increase if the length of the edges reach closer to their rest lengths. Hence, the maximum of  $P_i(z(0))$  is obtained when the initial positions of the edges satisfy

$$\|z_i(0) - z_j(0)\| = \text{sign}(\omega_{ij})(\|r_{ij}^*\| - l_{ij}) - \epsilon, \quad \forall (i, j) \in \mathcal{E}. \quad (33)$$

Consequently,

$$\begin{aligned} P_i(z(0))_{max} &= \sum_{\omega_{ij} < 0} \frac{k_{ij} [(l_{ij} - \|r_{ij}^*\|)^\alpha - \epsilon^\alpha]^\beta}{\epsilon} \\ &\quad + \sum_{\omega_{ij} > 0} \frac{k_{ij} [\|r_{ij}^*\| - l_{ij})^\alpha - \epsilon^\alpha]^\beta}{\epsilon}. \end{aligned} \quad (34)$$

Then, we let

$$P_{i\nu} = \frac{k_{i\nu} [(l_{i\nu} - \|r_{i\nu}^*\|)^\alpha - \rho_{i\nu}^\alpha]^\beta}{l_{i\nu} - \|r_{i\nu}^*\| - \|r_{i\nu} - r_{i\nu}^*\|} = P_i(z(0))_{max}. \quad (35)$$

So far, the  $\|r_{i\nu}\|$  can be derived from (34)-(35) theoretically. However, it can be seen that the resulting  $\|r_{i\nu}\|$  depends on the other edges' rest lengths and desired lengths, which is difficult to determine whether edge  $(i, \nu)$  reaches its rest length or not. Therefore, we further look for another relationship on the potential energy between the edge  $(i, \nu)$  and the other edges.

According to the statement above, to check whether  $\|r_{i\nu}\|$  reaches the rest length, only the potential energy of edge  $(i, \nu)$  increases, i.e., only the edge  $(i, \nu)$  changes towards its rest length, and the other edges move away from their rest lengths. This implies

$$0 < \rho_{i\nu}(t) \triangleq \epsilon_1 < \epsilon, \quad (36)$$

where  $\epsilon_1$  is a sufficiently small number. Now, we only take the numerators of (17) into consideration.

For two different edges, consider two functions

$$f_1(x) = k_{f1}(c_{f1}^\alpha - x^\alpha)^\beta, \quad k_{f1} > 0, \quad x \in (0, \epsilon], \quad (37)$$

and

$$f_2(x) = k_{f2}(c_{f2}^\alpha - x^\alpha)^\beta, \quad k_{f2} > 0, \quad x \in (0, \epsilon]. \quad (38)$$

Given sufficiently small number  $\epsilon_1$ , we have

$$f_1(\epsilon_1) = f_1(0) + \int_0^{\epsilon_1} f_1'(x)dx. \quad (39)$$

Since the function  $f_1(x)$  is continuous and differentiable in  $(0, \epsilon)$ , it follows from mean value theorem that

$$f_1(\epsilon_1) = f_1(0) + \epsilon_1 f_1'(s_1), \quad s_1 \in (0, \epsilon_1). \quad (40)$$

Similarly, for  $\epsilon > \epsilon_1$ , it follows

$$\begin{aligned} f_2(\epsilon) &= f_2(0) + \int_0^\epsilon f_2'(x)dx \\ &= f_2(0) + \int_0^{\epsilon_1} f_2'(x)dx + \int_{\epsilon_1}^\epsilon f_2'(x)dx \\ &= f_2(0) + \epsilon_1 f_2'(s_2) + (\epsilon - \epsilon_1) f_2'(s_3), \end{aligned} \quad (41)$$

where  $s_2 \in (0, \epsilon_1)$  and  $s_3 \in (\epsilon_1, \epsilon)$ .

From (13), the initial values satisfy

$$f_1(0) = f_2(0) = P_0. \quad (42)$$

And, from (23), for a sufficiently small  $\epsilon_1$ , under the condition that  $\alpha > 2$ , we have

$$f_1'(s_1) = f_2'(s_2) \rightarrow 0. \quad (43)$$

Considering the fact that  $f''(x) < 0$  in (24), it yields

$$f_2'(s_3) < f_2'(s_2) < 0. \quad (44)$$

Combining (40)-(44), we obtain

$$f_2(\epsilon) < f_1(\epsilon_1), \quad (45)$$

which implies

$$f_{ij}(\epsilon) < f_{i\nu}(\rho_{i\nu}), \quad j \in \mathcal{N}_i. \quad (46)$$

In view of (34)-(36), we know that the edge  $(i, \nu)$  satisfies

$$\begin{aligned} & k_{ij} [(\text{sign}(\omega_{ij})(\|r_{ij}^* - l_{ij}\|)^\alpha - \epsilon^\alpha)^\beta \\ & < k_{i\nu} [(l_{i\nu} - \|r_{i\nu}^*\|)^\alpha - \rho_{i\nu}(t)^\alpha]^\beta, \quad j \in \mathcal{N}_i. \end{aligned} \quad (47)$$

Taking (34)-(35) and (47) into consideration, we have

$$\frac{1}{l_{i\nu} - \|r_{i\nu}^*\| - \|r_{i\nu}(t) - r_{i\nu}^*\|} < \frac{|\mathcal{N}_i|}{\epsilon}, \quad (48)$$

where  $|\mathcal{N}_i|$  denotes the cardinality of the set  $\mathcal{N}_i$ , i.e., the number of agent  $i$ 's neighbors. Then, through simple calculation, we get

$$\|r_{i\nu}(t) - r_{i\nu}^*\| \leq l_{i\nu} - \|r_{i\nu}^*\| - \frac{\epsilon}{|\mathcal{N}_i|}. \quad (49)$$

Note that  $z_{i\nu}(t)$  satisfies

$$\|z_{i\nu}(t)\| = \|r_{i\nu}(t) - r_{i\nu}^*\| \geq \|r_{i\nu}(t)\| - \|r_{i\nu}^*\|. \quad (50)$$

Then, the upper bound for  $\|r_{i\nu}(t)\|$  can be derived from (49)-(50)

$$\|r_{i\nu}(t)\| \leq l_{i\nu} - \frac{\epsilon}{|\mathcal{N}_i|}, \quad (51)$$

which implies the length of edge  $(i, \nu)$  will not reach its rest length if it starts from  $\mathcal{D}^\epsilon(0)$ .

For the edge  $(i, \zeta)$  with positive stress, we can prove similarly that

$$\|r_{i\zeta}(t)\| \geq l_{i\zeta} + \frac{\epsilon}{|\mathcal{N}_i|}. \quad (52)$$

Therefore, we can draw the conclusion from (51) and (52) that the edges will never escape from the set  $\mathcal{D}(0)$  during the evolution, if the edges are initially located in  $\mathcal{D}^\epsilon(0)$ , namely, the set  $\mathcal{D}(0)$  is an invariant set. This completes the proof.  $\square$

**Remark 4.** Proposition 1 indicates that using the control law (21) the agents will not escape the set  $\mathcal{D}(0)$  if they start from  $\mathcal{D}^\epsilon(0)$ , which implies there will always be virtual forces along the edges. This further implies no sensing breakdown will happen since the virtual edges represent the sensing relationships between the agents.

Now, we are ready to present our main theorem.

**Theorem 3.** For any given initial position  $q(0) \in \mathcal{D}^\epsilon(0)$ , the formation stabilization problem of the networked single-integrator systems modeled by (10) can be solved using controller (21).

*Proof of Theorem 3.* From the control law (21) and potential function (17), we have

$$u_i = - \left( \sum_{\omega_{ij} < 0} \frac{\phi_{ij}}{\rho_{ij}^2 \|r_{ij} - r_{ij}^*\|} + \sum_{\omega_{ij} > 0} \frac{\varphi_{ij}}{\varrho_{ij}^2 \|r_{ij} - r_{ij}^*\|} \right) (z_i - z_j), \quad (53)$$

where

$$\begin{aligned} \phi_{ij} &= k_{ij} \left( \alpha \beta \rho_{ij}^\alpha [(l_{ij} - \|r_{ij}^*\|)^\alpha - \rho_{ij}^\alpha]^{\beta-1} + [(l_{ij} - \|r_{ij}^*\|)^\alpha - \rho_{ij}^\alpha]^\beta \right), \\ \varphi_{ij} &= k_{ij} \left( \alpha \beta \varrho_{ij}^\alpha [( \|r_{ij}^* \| - l_{ij} )^\alpha - \varrho_{ij}^\alpha]^{\beta-1} + [( \|r_{ij}^* \| - l_{ij} )^\alpha - \varrho_{ij}^\alpha]^\beta \right), \end{aligned}$$

with  $\rho_{ij}$  and  $\varrho_{ij}$  being defined in (18). We further write the closed-loop system (10) and (17) into their compact form

$$\dot{\bar{q}} = \dot{z} = -(\mathcal{L}_w(l, \bar{q}, \bar{q}^*) \otimes I_d)z, \quad (54)$$

where  $\bar{q} = [q_1^T, \dots, q_n^T]^T \in \mathbb{R}^{dn}$ ,  $\bar{q}^* = [(q_1^*)^T, \dots, (q_n^*)^T]^T \in \mathbb{R}^{dn}$ , and  $\mathcal{L}_w = H^T W(l, \bar{q}, \bar{q}^*) H$  is  $\mathcal{G}$ 's weighted Laplacian matrix.

$H$  is  $\mathcal{G}$ 's incidence matrix defined in (1), and the diagonal weight matrix  $W(l, \bar{q}, \bar{q}^*)$  is

$$W(l, \bar{q}, \bar{q}^*) = \text{diag} \{w_i\}, \quad i = 1, 2, \dots, |\mathcal{E}|, \quad \text{with} \quad (55)$$

$$w_i = \begin{cases} \frac{\phi_i}{\rho_i^2 \|r_i - r_i^*\|}, & \text{if } \omega_i < 0, \\ \frac{\varphi_i}{\varrho_i^2 \|r_i - r_i^*\|}, & \text{if } \omega_i > 0. \end{cases}$$

Given  $q(0) \in \mathcal{D}^\epsilon(0)$ , it follows from Proposition 1 that  $\mathcal{L}_w$  in (54) is well defined and positive semi-definite [27]. Also, the interaction graph of the agents is connected due to the fact that the edges are always in tension or in compression. For an undirected connected graph, it is well-known that  $\mathbf{1}_n$  is an eigenvector of the Laplacian matrix associated with the simple zero eigenvalue 0 [28]. Then, we have

$$\lim_{t \rightarrow \infty} z_1(t) = \lim_{t \rightarrow \infty} z_2(t) = \dots = \lim_{t \rightarrow \infty} z_n(t), \quad (56)$$

which implies that

$$\lim_{t \rightarrow \infty} r_{ij}(t) = r_{ij}^*, \quad \forall (i, j) \in \mathcal{E}. \quad (57)$$

Thus, the prescribed formation is achieved. This completes the proof.  $\square$

**Remark 5.** Even though the result is in the sense of local stability, we can enlarge the stability region by choosing small  $\gamma_{ij}^\epsilon$  for edges with positive stress and large  $\gamma_{ij}^\epsilon$  for those with negative stress. The definition of  $\mathcal{D}^\epsilon(0)$  also describes clearly an estimate of the domain of attraction of the desired equilibrium of the closed-loop system.

**Remark 6.** Note that the configuration of the desired formation coincides with the given configuration  $q^*$  in (3) for designing universally rigid tensegrity framework, we thus can stabilize the proper formation consisting of any arbitrary number of agents theoretically. When one or more agents encounter mechanical failures, to stabilize the rest, the weights  $k_{ij}$  in control law (53) are required to be updated by recalculating the stresses  $\omega_{ij}$  based on the proposed algorithm, where the matrix  $Q^*$  needs to be altered via removing the failure agents' configuration. The resultant formation will remain the same as the original one associated with the agents without failure.

**Remark 7.** From (57), we know the equilibrium configuration is

$$\lim_{t \rightarrow \infty} q(t) = q^* + \gamma \text{span}(\mathbf{1}_d),$$

where  $\gamma$  is an arbitrary real number. Indeed, the stabilized configuration  $q$  is a class of affine transformation of  $q^*$  [29], which is still in equilibrium with the stress  $\omega$ , i.e.,

$$\sum_{j \in \mathcal{N}_i} \omega_{ij} (q_i - q_j) = \sum_{j \in \mathcal{N}_i} \omega_{ij} (q_i^* - q_j^*) = \mathbf{0}, \quad i = 1, \dots, n.$$

**Remark 8.** In this paper, the single integrator model is taken into account under the assumption that the velocities of the agents can be controlled directly. In practice, the signals generated by the system can serve as the commanded 3D-velocity for the tracking controllers of the quadrotors [30, 31] or vessels [32]. This technique has been employed in experimental setups in formation and motion control [33].

## V. SIMULATION RESULTS

This section gives the simulation results to validate the effectiveness of the theoretical results derived in the preceding sections. For the construction of universally rigid tensegrity frameworks, we consider a general as well as generic configuration. Then, the formation stabilization algorithm is simulated on the resultant universally rigid tensegrity framework.

### A. Construction of universally rigid tensegrity frameworks

The generic configuration is given as follows

$$q^* = \begin{bmatrix} -\sqrt{3}/2 & \sqrt{3}/2 & 0 & -1/2 & 1 & -2/3 \\ -1/2 & -1/2 & 1 & -1/2 & 0 & 8/3 - \sqrt{3} \\ 0 & 0 & 0 & 3 & 3 & 3\sqrt{3} - 2 \end{bmatrix}. \quad (58)$$

With this configuration, the corresponding geometric shape of the six nodes is shown in Fig. 1. Based on our proposed algorithm, one solution of the Gale matrix  $D$  of the Gale matrix  $D$  derived from (7) is given as follows

$$D = \frac{1}{3} \begin{bmatrix} 2\sqrt{3} & -\sqrt{3} & -\sqrt{3} & -3 & 3 & 0 \\ 0 & 3 & 3\sqrt{3} - 8 & -\sqrt{3} & 2 - 2\sqrt{3} & 3 \end{bmatrix}^T.$$

Then the stress matrix  $\Omega$  can be calculated via  $\Omega = DD^T$ :

$$\Omega = \begin{bmatrix} 12 & -6 & -6 & -6\sqrt{3} & 6\sqrt{3} & 0 \\ -6 & 12 & 9\sqrt{3} - 21 & 0 & 6 - 9\sqrt{3} & 9 \\ -6 & 9\sqrt{3} - 21 & 94 - 48\sqrt{3} & 11\sqrt{3} - 9 & 19\sqrt{3} - 34 & 9\sqrt{3} - 24 \\ -6\sqrt{3} & 0 & 11\sqrt{3} - 9 & 12 & -3 - 2\sqrt{3} & -3\sqrt{3} \\ 6\sqrt{3} & 6 - 9\sqrt{3} & 19\sqrt{3} - 34 & -3 - 2\sqrt{3} & 25 - 8\sqrt{3} & 6 - 6\sqrt{3} \\ 0 & 9 & 9\sqrt{3} - 24 & -3\sqrt{3} & 6 - 6\sqrt{3} & 9 \end{bmatrix}. \quad (59)$$

It can be seen from  $\Omega$  that 13 members are required to construct a universally rigid tensegrity framework, 10 of which are cables and 3 struts. The corresponding tensegrity framework is shown in Fig. 2, where the thin black and thick blue lines are cables and struts, respectively. It is worth noting that 13 is exactly the minimal number of members required to build a universally rigid tensegrity framework in generic configurations with 6 nodes in 3D.

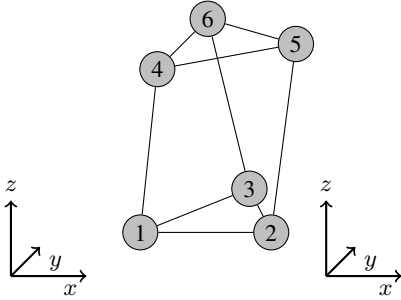


Fig. 1. Desired geometric shape associated with (58).

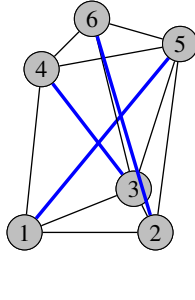


Fig. 2. Resultant framework based on  $\Omega$  in (59).

### B. Formation stabilization

Base on the obtained tensegrity framework, where the nodes represent the agents modeled by (10), the control input (53) for each agent can be specified. For simplification, we set the parameters  $\gamma_{ij}^s$  to be the same value  $\gamma^s = 2$ , and  $\gamma_{ij}^c$  is chosen to be  $\gamma^c = 0.5$ . The initial states of  $q(t)$  are chosen as  $q_i(0) = q_i^* + 0.3 * \text{rands}(3, 1)$ ,  $i = 1, 2, \dots, 6$ .

It can be seen from Fig. 3 that the length of each member converges to its desired one. The upper panel of Fig. 3 shows the stabilization errors of the cables, and the corresponding stabilization errors of the struts are shown in the lower panel. The length evolution intervals of the cables and struts together with their rest lengths are presented respectively in Table I and II, from which we can observe that the length of any cable (resp. strut) is always longer (resp. shorter) than its rest length during the evolution, verifying Proposition 1 numerically. As can be seen from Fig. 3, all the stabilization errors converge to zero within 0.1s. Then the formation shape variations during  $[0, 0.1]$ s

are depicted sequentially in Fig. 4, where the motion of the straight line of the whole formation results from an additional control input  $u_e = [140, 0, 0]^T$  for each agent. This input is independent of the system state, and thus can be separated from the control input (53). The intention of designing  $u_e$  is to clearly show the variation of the geometric shape of the formation during the system evolution. Overall, these numerical results indicate that the prescribed formation can be achieved using the virtual framework and our proposed control algorithm (53).

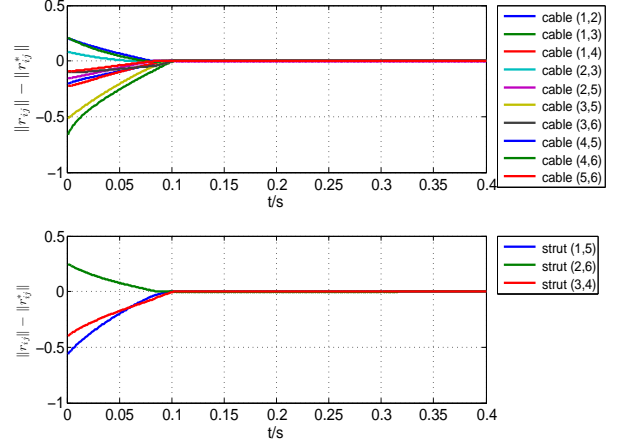


Fig. 3. The stabilization errors of the members.

TABLE I  
LENGTH EVOLUTION INTERVALS OF THE CABLES AND THEIR REST LENGTHS.

Cable	Length interval	Rest length
(1, 2)	[1.5238, 1.7328]	0.8660
(1, 3)	[1.0683, 1.7340]	0.8660
(1, 4)	[2.8182, 3.0474]	1.5222
(2, 3)	[1.7211, 1.8152]	0.8600
(2, 5)	[2.8860, 3.0471]	1.5222
(3, 5)	[2.8011, 3.3172]	1.6583
(3, 6)	[3.1644, 3.2659]	1.6328
(4, 5)	[1.7305, 1.9374]	0.8600
(4, 6)	[1.8143, 2.0299]	0.9095
(5, 6)	[1.8263, 1.9219]	0.9604

TABLE II  
LENGTH EVOLUTION INTERVALS OF THE STRUTS AND THEIR REST LENGTHS.

Strut	Length interval	Rest length
(1, 5)	[3.0050, 3.5694]	7.1364
(2, 6)	[3.8210, 4.0727]	7.6479
(3, 4)	[3.1661, 3.5706]	7.1364

## VI. CONCLUSION

In this paper, we have proposed a numerical algorithm, based on which universally rigid tensegrity frameworks can be built for any given configurations in general position. Furthermore, the upper bound of the members inserted in the framework has also been given.



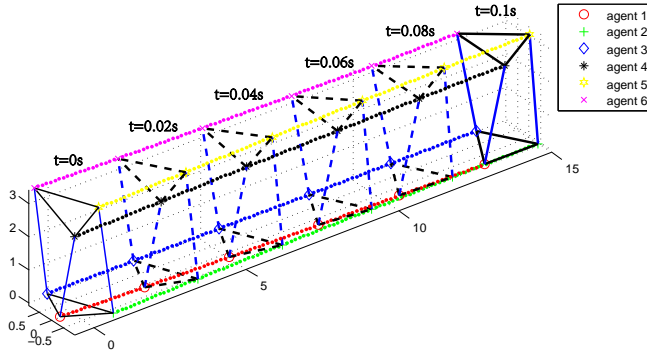


Fig. 4. Formation evolution between  $t = 0 - 0.1s$ .

Then, we have investigated the formation stabilization problem as one of the applications, where distributed control strategies have been designed, such that the prescribed formation can be realized. During the stabilization evolution, the lengths of the members are shown to vary over or below some bounds, which is of great interest for tethered robot control that will be tested using real robots in our lab.

#### REFERENCES

- [1] F. Dorfler and B. Francis. Geometric analysis of the formation problem for autonomous robots. *IEEE Transactions on Automatic Control*, 55(10):2379–2384, 2010.
- [2] M. Cao, C. Yu, and B. Anderson. Formation control using range-only measurements. *Automatica*, 47(4):776–781, 2011.
- [3] Z. Lin, L. Wang, Z. Chen, M. Fu, and Z. Han. Necessary and sufficient graphical conditions necessary and sufficient graphical conditions for affine formation control. *IEEE Transactions on Automatic Control*, 61(10):2877–2891, 2016.
- [4] W. Whiteley. Rigidity of molecular structures: generic and geometric analysis. In *Rigidity Theory and Applications*, pages 21–46. Springer, 2002.
- [5] Z. Zhu, A. M. C. So, and Y. Ye. Universal rigidity: Towards accurate and efficient localization of wireless networks. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9. IEEE, 2010.
- [6] J. B. Saxe. *Embeddability of weighted graphs in k-space is strongly NP-hard*. Carnegie-Mellon University, Department of Computer Science, 1980.
- [7] R. Connelly. Tensegrity structures: why are they stable? In *Rigidity theory and applications*, pages 47–54. Plenum Press, New York, 1999.
- [8] R. Connelly. Generic global rigidity. *Discrete & Computational Geometry*, 33:549–563, 2005.
- [9] A. Y. Alfakih. On the universal rigidity of generic bar frameworks. *Contributions to Discrete Mathematics*, 5(1), 2010.
- [10] S. J. Gortler and D. P. Thurston. Characterizing the universal rigidity of generic frameworks. *Discrete & Computational Geometry*, 51(4):1017–1036, 2014.
- [11] R. Connelly and S. J. Gortler. Iterative universal rigidity. *Discrete & Computational Geometry*, 53(4):847–877, 2015.
- [12] A. Y. Alfakih, N. Taheri, and Y. Ye. On stress matrices of  $(d+1)$ -lateration frameworks in general position. *Mathematical Programming*, 137(1-2):1–17, 2013.
- [13] A. Y. Alfakih and Y. Ye. On affine motions and bar frameworks in general position. *Linear Algebra and its Applications*, 438(1):31–36, 2013.
- [14] T. Jordán and V. H. Nguyen. On universally rigid frameworks on the line. *Contributions to Discrete Mathematics*, 10(2), 2016.
- [15] R. Connelly and W. Whiteley. Second-order rigidity and prestress stability for tensegrity frameworks. *SIAM Journal on Discrete Mathematics*, 9(3):453–491, 1996.
- [16] J. M. M. Tur and S. H. Juan. Tensegrity frameworks: dynamic analysis review and open problems. *Mechanism and Machine Theory*, 44(1):1–18, 2009.
- [17] T. Liedl, B. Högberg, J. Tytell, D. E. Ingber, and W. M. Shih. Self-assembly of three-dimensional prestressed tensegrity structures from dna. *Nature nanotechnology*, 5(7):520–524, 2010.
- [18] R. Connelly. Rigidity and energy. *Inventiones Mathematicae*, 66(1):11–33, 1982.
- [19] S. D. Kelly and A. Micheletti. A class of minimal generically universally rigid frameworks. *arXiv preprint arXiv:1412.3436*, 2014.
- [20] B. Nabet and N. E. Leonard. Tensegrity models and shape control of vehicle formations. *arXiv: 0902.3710*, 2009.
- [21] D. Pais, M. Cao, and N. E. Leonard. Formation shape and orientation control using projected collinear tensegrity structures. In *Proceedings of the 2009 American Control Conference*, pages 610–615, St. Louis, MO, June 2009.
- [22] S. Lau and W. Naeem. Tensegrity-based formation control of unmanned vehicles. In *Control (CONTROL), 2012 UKACC International Conference on*, pages 1–6. IEEE, 2012.
- [23] S. Lau and W. Naeem. Co-operative tensegrity-based formation control algorithm for a multi-aircraft system. In *2015 American Control Conference (ACC)*, pages 750–756. IEEE, 2015.
- [24] A. Pothén. *Sparse null bases and marriage theorems*. PhD thesis, Cornell University, 1984.
- [25] L. Gottlieb and T. Neylon. Matrix sparsification and the sparse null space problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 205–218. Springer, 2010.
- [26] John R. Gilbert and Michael T. Heath. Computing a sparse basis for the null space. *SIAM Journal on Algebraic Discrete Methods*, 8(3):446–459, 1987.
- [27] M. Ji and M. Egerstedt. Distributed coordination control of multi-agent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007.
- [28] W. Ren and R. Beard. *Distributed consensus in multi-vehicle cooperative control: theory and applications*. Springer-Verlag, 2008.
- [29] R. Connelly. Tensegrities and global rigidity. In *Shaping Space*, pages 267–278. Springer, 2013.
- [30] G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin. Quadrotor helicopter trajectory tracking control. In *AIAA guidance, navigation and control conference and exhibit*, pages 1–14, 2008.
- [31] D. Mellinger, N. Michael, and V. Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, page 0278364911434236, 2012.
- [32] T. I. Fossen. *Guidance and control of ocean vehicles*. John Wiley & Sons Inc, 1994.
- [33] J. Fink, N. Michael, S. Kim, and V. Kumar. Planning and control for cooperative manipulation and transportation with aerial robots. *The International Journal of Robotics Research*, 30(3):324–334, 2011.